Lab 1

Deadline: As specified in Moodle in the week of Jan 22, 2024

Head over to OnlineGDB, where you'll set up your Python environment and write your Python programs.

Tool Setup

Python is a strong and flexible programming language that can be used for a wide range of tasks, including artificial intelligence, data analysis, and web development. This lab will walk you through the basic steps of Python and assist you in setting up the resources you'll need to begin programming.

Program 0

1. **OnlineGDB Platform**

We will be using OnlineGDB for this class, a convenient online Python compiler and debugger.

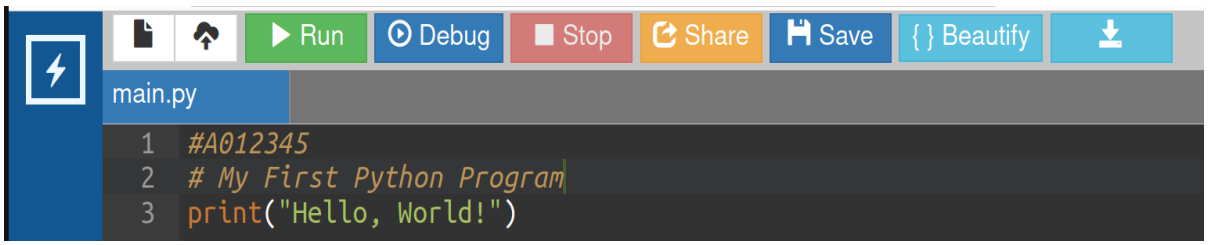
- Open your web browser and navigate to OnlineGDB .
- If you don't have an account, you can sign up for free. If you already have an account, log in.

2. **Create New Project**

- Once you are logged in, click on "Create New Project" to create a new Python project.
- Choose "Python 3" as the language for your project.
- Click the “Save” button to save your project as Lab1.

3. **Writing Your First Python Program**

- In the code editor, type the following Python program:



- Click the “Run” button to execute the program. You should see the output "Hello, World!" in the console.

4. **Save the File Offline**

Click the “Download Code” button/icon to save the file offline. Rename the file to q0.py.

Program 1

A company had a busy business week and wants to reward its employees with a bonus based on their hours worked and employee category. There are three categories of employees: assistants (A), managers (M), and executives (E). The company has specific hourly rates for each category and additional bonus based on the hours worked.

1. **Requirements**

Write a Python program to calculate the total salary including bonus e.g total_salary= salary+bonus for an employee based on the following criteria:

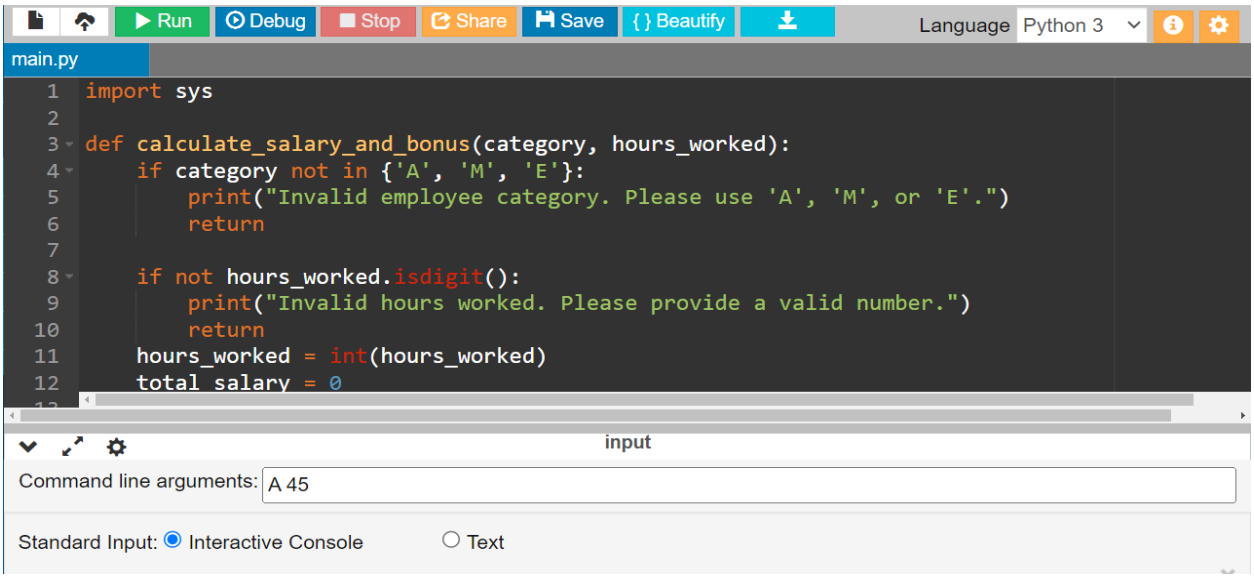
- Assistants (A) have an hourly rate of $25 for the first 40 hours per week. For hours beyond 40, add $5 per hour to the rate. If an assistant worked between 41-50 hours, add a bonus of $100; or, if the hours are between 51-60, double the bonus.
- Managers (M) have an hourly rate of $40 for the first 40 hours per week. For hours beyond 40, add $10 per hour to the rate. If a manager worked between 41-50 hours, add a bonus of $150; or, if the hours are between 51-60, double the bonus.
- Executives (E) have an hourly rate of $75 for the first 40 hours per week. For hours beyond 40, add $25 per hour to the rate. If an executive worked between 41-50 hours, add a bonus of $250; or, if the hours are between 51-60, double the bonus.

The program reads employee category (A, M, or E) and the number of hours worked as command line arguments. Calculate the total salary including bonus based on the given criteria and display the result.

2. **How to do**

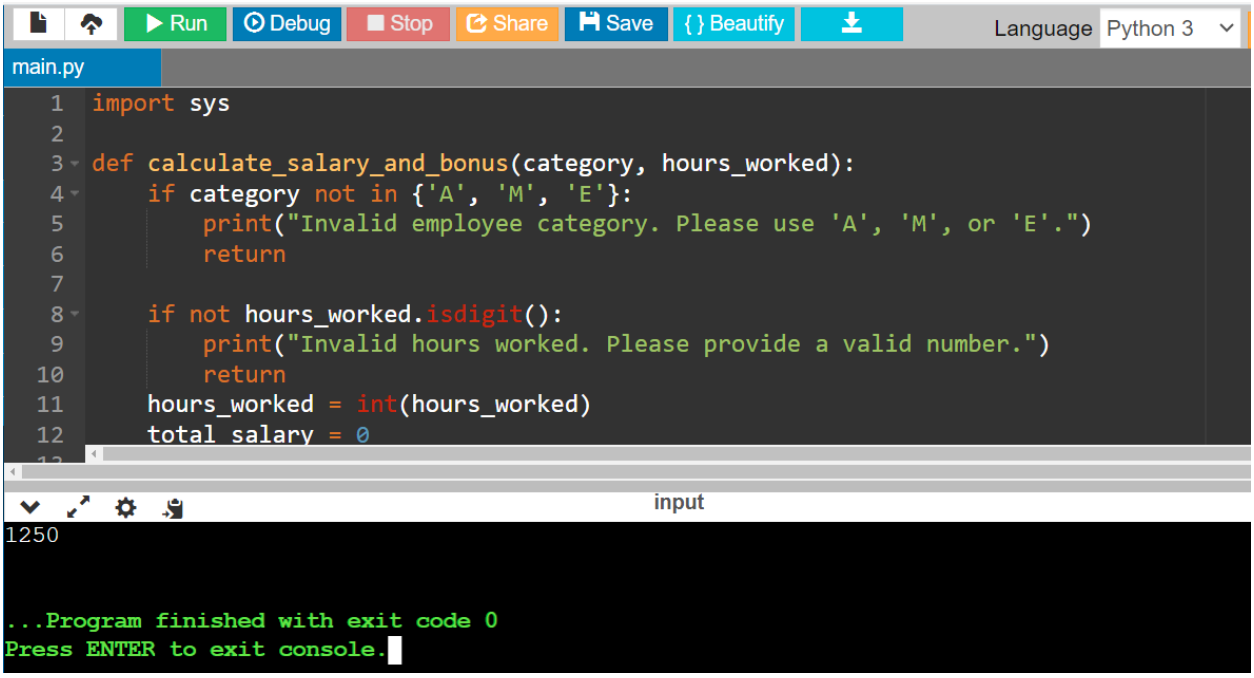
You are provided with the q1.py skeleton file. Upload the file to OnlineGDB first and then select Python 3 as a language. Write your code from “#Write your code here” line. You are not supposed to change any other line of code in skeleton file.

How to run your code



Add command line arguments as input e.g A 45 and click Run.

3. **Output**



4. **Save the File Offline**

Click the “Download Code” button/icon to save the file offline. Rename the file to q1.py.

Program 2

A trader bought a stock of goods consisting of 1000 items and sold it. After selling the stock, the price increased by $2 per item, and the trader had to restock the goods. Write a Python program to calculate the net profit or loss of the trader after restocking.

1. **Requirements**

Your program should:

- Read the buying price per item and selling price per item as command line arguments e.g 10 15.
- Calculate the cost of restocking at the increased price.
- Calculate the net profit or loss by subtracting the restocking cost from total selling price.
- Display the result.

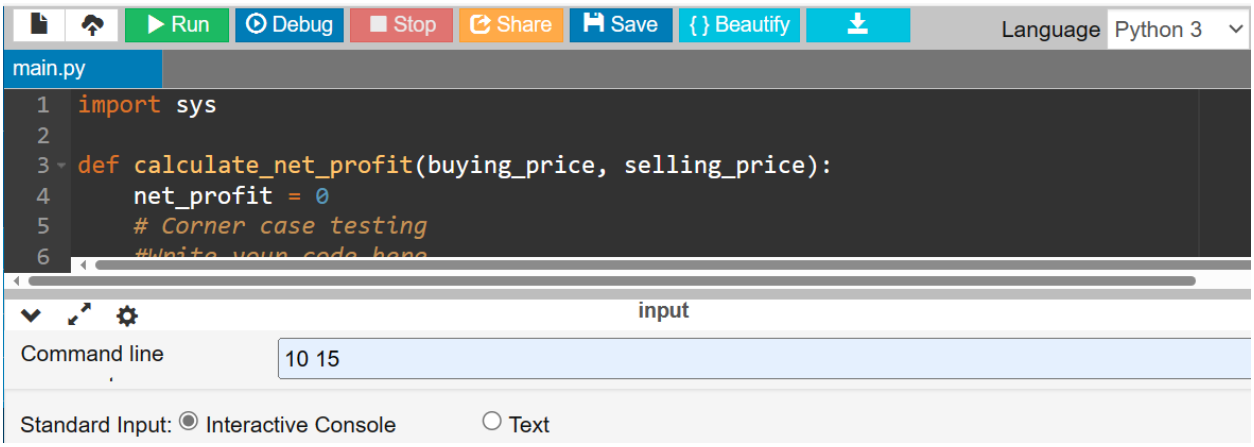
2. **Test and Validation**

This question challenges to handle input validation for positive values and implement corner testing to handle potential edge cases e.g a character input as price.

3. **How to do**

You are provided with the q2.py skeleton file. Reopen OnlineGDB in new tab. Upload the file to OnlineGDB first and then select Python 3 as a language. Write your code from "#Write your code here" line. You are not supposed to change any other line of code in skeleton file.

4. **How to run your code**



Add command line arguments as input e.g 10 15 and click Run

5. **Output**

The program should print the amount of net profit/loss only or appropriate error message.

6. **Save the File Offline**

Click the "Download Code" button/icon to save the file offline. Rename the file to q2.py.

Submission

- Include your student number as comment at the top of your code in all files i.e #012345.
- Make sure that each py file only contains python code for the particular question. For example, program 2 code should only be in q2.py, not in q1.py.
- Place q0, q1, and q2 in a new folder. Compress the folder as a zip file and upload to moodle.

Grading

- q0.py: 1 pt
- q1.py: 2 pts
- q2.py: 2 pts